

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A connection scheduling method, operable in a node of a networked data processing system comprising a plurality of nodes, comprising the steps of:
 - determining if a job is available for scheduling;
 - determining, in response to said step of determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;
 - creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;
 - launching said session to effect said execution of said job, if said session is available; and
 - launching an error handling thread in response to an error condition, said error handling thread releasing said session.
2. (Previously Presented) The method of claim 1 wherein said session comprises a thread.
3. (Cancelled)
4. (Currently Amended) The method of claim [[3]] 1 further comprising the step of determining if said network connection is an existing network connection, and wherein said step of creating said network connection is performed if said network connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions.
5. (Cancelled)
6. (Previously Presented) The method of claim 1 further comprising the step of changing value of a job state from a first value to a second value in response to said launching of said error handling thread.

7. (Original) The method of claim 6 wherein said first value signals that said job is available for scheduling.
8. (Previously Presented) The method of claim 1 further comprising the step of retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.
9. (Original) The method of claim 8 wherein said step of retrying is repeated until a predetermined time interval has elapsed.
10. (Original) The method of claim 9 further comprising the step of registering a callback method in response to an expiry of said predetermined time interval.
11. (Previously Presented) The method of claim 10 wherein said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session are performed in response to an invoking of said callback method by said target system.
12. (Previously Presented) A data processing system for connection scheduling within a network comprising a plurality of nodes, comprising:
circuitry operable for determining if a job is available for scheduling;
circuitry operable for determining, in response to said circuitry operable for determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;
circuitry operable for creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;
circuitry operable for launching said session to effect said execution of said job, if said session is available; and
circuitry operable for launching an error handling thread in response to an error condition, said error handling thread releasing said session.
13. (Previously Presented) The system of claim 12 wherein said session comprises a thread.
14. (Cancelled)

15. (Currently Amended) The system of claim [[14]] 12 further comprising circuitry operable for determining if said network connection is an existing network connection, and wherein said circuitry operable for creating said network connection is operated if said network connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions.
16. (Cancelled)
17. (Previously Presented) The system of claim 12 further comprising circuitry operable for changing value of a job state from a first value to a second value in response to said launching of said error handling thread.
18. (Original) The system of claim 17 wherein said first value signals that said job is available for scheduling.
19. (Previously Presented) The system of claim 12 further comprising circuitry operable for retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.
20. (Original) The system of claim 19 wherein said circuitry operable for retrying is operated until a predetermined time interval has elapsed.
21. (Original) The system of claim 20 further comprising circuitry operable for registering a callback method in response to an expiry of said predetermined time interval.
22. (Previously Presented) The system of claim 21 wherein said circuitry operable for determining if a job is available for scheduling, determining if a session is available, and launching said session are operated in response to an invoking of said callback method by said target system.
23. (Previously Presented) A computer program product embodied in a machine readable storage medium, the program product for job scheduling comprising instructions for:
determining if a job is available for scheduling;

determining, in response to instructions for determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;

creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;

launching said session to effect said execution of said job, if said session is available; and

launching an error handling thread in response to an error condition, said error handling thread releasing said session.

24. (Previously Presented) The program product of claim 23 wherein said session comprises a thread.

25. (Cancelled)

26. (Currently Amended) The program product of claim [[25]] 23 further comprising instructions for determining if said network connection is an existing network connection, and wherein said instructions for creating said network connection are performed if said connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions.

27. (Cancelled)

28. (Previously Presented) The program product of claim 23 further comprising instructions for changing value of a job state from a first value to a second value in response to said launching of said error handling thread.

29. (Original) The program product of claim 28 wherein said first value signals that said job is available for scheduling.

30. (Previously Presented) The program product of claim 29 further comprising programming for retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.

31. (Original) The program product of claim 30 wherein said instructions for retrying are repeated until a predetermined time interval has elapsed.
32. (Original) The program product of claim 31 further comprising programming for registering a callback method in response to an expiry of said predetermined time interval.
33. (Previously Presented) The program product of claim 32 wherein said instructions for determining if a job is available for scheduling, determining if a session is available, and launching said session are executed in response to an invoking of said callback method by said target system.